REMARKS

Claims 1-24 and 26-31 are pending. Of these, claims 1-3, 6-14, 17-24 and 28-31 are rejected, while claims 4, 5, 15, 16, 26 and 27 are objected to as being dependent from a rejected base claim. By the above amendments, claims 3, 14 and 26 are amended. The applicants request further examination and consideration in view of the amendments above and remarks set forth below.

Claims 3, 14 and 26 are amended to improve their clarity. Particularly, claims 3 and 14 are amended to recite a further step of reducing one or more of the performance parameters from said desired levels. Claim 26 is amended to recite that one or more of the performance parameters is reduced from said desired levels. No new matter has been added.

Claim Rejections:

Claims 1-3, 6-14, 17-24 and 28-31 are rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 6,487,562 to Mason, Jr. et al. (hereinafter "Mason") in view of U.S. Patent No. 5,345,579 to Hynes (hereinafter "Hynes").

Regarding claim 1, the office action states that Mason teaches all of its limitations except for the step of "predicting levels of performance parameters for the modified design." However, the office action states that Hynes teaches this feature and that it would have been obvious to modify Mason with the performance prediction of Hynes because:

> There are several advantages of using the fixed class workloads instead of the transaction workloads. First, the fixed class workloads provide better approximations to the terminal and batch workloads than the transaction workloads. Second, the fixed class workloads do not distort the performance metrics for those terminal and batch workloads which are specified using population values. Third, the fixed class workloads yield the actual fixed class population sizes.

Quoting from Hynes at col. 3, lines 30-39.

The applicants respectfully traverse the rejection at least because the Mason and Hynes references are not properly combinable. First, the proposed modification to Mason would negate the clear teachings of Mason that changes are to be made dynamically to the data storage system of Mason while it is in use. Second, the alleged motivation in Hynes to make the combination (quoted above) relates to the type of models used by Hynes for characterizing workloads. These workload models

are completely unrelated to the system of Mason and, therefore, advantages of these models cannot provide a motivation for modifying Mason. These reasons are discussed more fully below.

Regarding the first reason (that the proposed modification to Mason would negate the clear teachings of Mason), Mason points out that a drawback with prior data storage systems is that it is difficult and time consuming to attempt to "tweak" or tune system performance because users are not able to make changes to the system behavior while the system is running. Mason at col. 1, lines 57-63. To solve this problem, Mason discloses a system and method for dynamically modifying parameters in a data storage system. Mason at col. 1, lines 66-67. Mason teaches that quality of service functions are controlled through the user interface while the data storage system is running. Mason at col. 2, lines 38-39. Thus, advantages espoused by Mason include the ability to make changes dynamically to the data storage system while it is in use. Mason at col. 2, lines 50-52. Particularly, the changes made to the QOS of various system parameters are said to take effect quickly enough to be almost transparent to the system administrator, and to end users. Mason at col. 2, lines 52-55. This is said to allow system administrators to observe performance changes in real-time, and thereby optimize the system through immediate feedback. Mason at col. 2, lines 55-57.

Therefore, based on these clear teachings of Mason, a person of ordinary skill in the art would not have been motivated to make the proposed modification to Mason, in which the model solution software module of Hynes is used to generate predicted performance metrics, because such a modification would negate the clear teachings of Mason that changes to parameters are to be made in real-time to the data storage system while it is in use. Therefore, it would not have been obvious to make the combination of Mason and Hynes. For at least this reason, claims 1-3, 6-14, 17-24 and 28-31 are allowable over Mason and Hynes.

Regarding the second reason (that the alleged motivation to make the combination relates to types of workload models which are completely unrelated the system of Mason), Hynes discloses a computer software program for solving models of computer systems. Hynes at col. 3, lines 6-8. Hynes explains that computer system models are used for predicting the manner in which computer systems will react under future conditions and loads. Hynes at col. 1, lines 10-14. The models are developed by characterizing workloads which fall into three classes: a terminal class,

a batch class and a transaction class. Hynes at col. 1, lines 18-23. Terminal and batch classes represent closed classes which have finite populations while transaction classes represent open classes which have infinite populations. Hynes at col. 1, lines 24-37. The workload classes are specified using measured data. Hynes at col. 1, lines 38-53.

Hynes explains that model solutions are algorithms which receive as input the computer system models and which generate performance metrics of the computer systems. Hynes at col. 1, lines 54-56. A particular known model solution algorithm is often used when the models include only closed workloads and the available measured information includes the population. Hynes at col. 1, line 63 to col. 2, line 5. However, the population size is often unavailable and therefore that model solution cannot be used. Hynes at col. 2, lines 6-24. A prior solution to this problem is to replace the terminal and batch workloads with transaction workloads. Hynes at col. 2, lines 25-34. However, this solution results often fails to validate the model due to error inherent in representing closed workloads with open workloads. Hynes at col. 2, lines 35-49.

To overcome these disadvantages, the algorithm disclosed by Hynes uses fixed class workloads and implements a fixed class algorithm as the model solution. Hynes at col. 3, lines 12-19. This is said to provide the following advantages: first, the fixed class workloads provide better approximations to the terminal and batch workloads than the transaction workloads; second, the fixed class workloads do not distort the performance metrics for those terminal and batch workloads which are specified using population values; and third, the fixed class workloads yield the actual fixed class population sizes. Hynes at col. 3, lines 31-39.

Therefore, these advantages espoused by Hynes are clearly specific to the workload models that are used with the model solution algorithm of Hynes. However, because Mason does not use any models or model solution algorithms, there would be no need to consider the type of workload models to employ in the system of Mason. Accordingly, the purported advantages of a particular workload model type could not have provided a motivation to combine Hynes with Mason. This is another reason why claims 1-3, 6-14, 17-24 and 28-31 are allowable over Mason and Hynes.

Moreover, at least claims 10-14, 17-24 and 28-31 include limitations that are not taught by Mason or Hynes. Therefore, these claims are allowable even if Mason

and Hynes references could be combined. More particularly, claim 10 is an independent claim that recites predicting levels of performance parameters for the design, comparing the predicted levels of performance parameters to the desired levels of performance parameters and modifying the design including modifying the assignments of the system resources when the predicted levels are lower than the desired levels. Claim 23 is an independent claim that recites a program loop in which performance parameter levels are predicted for the design, the predicted performance parameters are compared to the desired levels of performance parameters and the design is modified, including modifying assignments of system resources to applications in response to the comparison. The applicants submit that the Mason and Hynes references, taken singly or in combination, do not teach or suggest these features of claims 10 and 23.

Particularly, Mason teaches a system and method for dynamically modifying parameters in a data storage system. Mason at col. 1, lines 66-67. Mason teaches that control is allowed over "certain provided services on a logical volume basis." Mason at col. 2, lines 5-7. The services fall into three categories: data replication/recreation through a copy mechanism, performance management though control of caching services, and data integrity checks. Mason at col. 2, lines 7-11 and col. 6, lines 6-12. A QOS data structure is associated with each logical volume. Mason at col. 6, lines 33-39. The QOS data structure stores priority levels for the copy services and the scrub services (i.e. data integrity checks) and a cache services selection bitmap. Table 1 of Mason at col. 6, lines 40-60. The cache services selection bitmap controls cache policies such as prefetch and LRU algorithms. Table 2 of Mason at col. 9, lines 1-24. When a parameter in the QOS data structure is changed, a storage device controller 19 reconfigures the QOS for the logical volume. Mason at col. 7, lines 3-6. If the system administrator attempts to set any parameters that are not valid, for example, parameters that are contradictory or that create a tautology, an error is signaled. Mason at col. 9, lines 43-47.

However, Mason does not teach comparing predicted levels of performance parameters to desired levels, as is required by applicants' claims 10 and 23. This is clear because Mason does not perform any modeling or have other functionality that could predict performance. Moreover, because Mason does not compare predicted levels of performance parameters to desired levels, Mason also does not modify

assignments of system resources based on such a comparison, as is also required by claims 10 and 23.

Hynes discloses a computer software program for solving models of computer systems. Hynes at col. 3, lines 6-8. Hynes explains that computer system models are used for predicting the manner in which computer systems will react under future conditions and loads. Hynes at col. 1, lines 10-14. Figure 4 of Hynes illustrates a functional flowchart of its model solution application computer software program. In steps 402 and 410, a user measures the performance of the computer system and classifies the computer system's workloads as either fixed or non-fixed. Hynes at col. 5, lines 33-39. As a result of steps 402 and 410, model solution inputs 414 are produced which represent the model of the computer system. Hynes at col. 5, lines 40-50. In step 418, the model solution module receives the model solution inputs 414 and generates performance metrics 422 of the computer system model which represent the manner in which the computer system would react under the conditions and loads which are specified in the model solution inputs 414. Hynes at col. 5, lines 51-57. The output module 206 is used to display the performance metrics 422 to the user. Hynes at col. 5, lines 57-58. In step 426, the performance metrics 422 and measured data 406 are compared. Hynes at col. 5, lines 61-62. An unfavorable comparison indicates that the model solution inputs 414 do not represent an accurate model of the computer system. Hynes at col. 5, lines 62-64. A loop which represents a validation step is performed until the comparison is satisfactory. Hynes at col. 5, line 65 to col. 6, line 5. In step 436, the user inputs new model solution inputs 440 to reflect new computer system conditions and loads. Hynes at col. 6, lines 6-10. In step 444, the model solution module receives the model solution inputs 440 and generates performance metrics which represent the manner in which the computer system would react to the conditions and loading as specified in the model solution inputs 440. Hynes at col. 6, lines 11-16. In step 452, the output module 206 displays the performance metrics 448. Hynes at col. 6, lines 17-18.

While Hynes discloses a software program for solving models of computer systems, Hynes does not teach or suggest comparing predicted levels of performance parameters to desired levels, nor does Hynes teach or suggest modifying assignments of system resources based on such a comparison, as is required by claims 10 and 23. Though Hynes makes a comparison in step 426, this step compares outputs of the model to measurements. This comparison is made by Hynes to determine whether the

model solution inputs represent an accurate model of the computer system. In contrast, the comparison performed according to claims 10 and 23 is between predicted levels of performance parameters and desired levels. And, according to claims 10 and 23, this comparison is made in order to determine how to modify assignments of system resources to applications. Therefore, differences between claims 10 and 23 and Hynes include different parameters being compared for different reasons. And, different actions are performed based on results of the comparisons.

Because neither Mason, nor Hynes, taken singly or in combination, discloses all of their limitations, claims 10 and 23 are allowable over Mason and Hynes. Claims 11-22 are also allowable at least because they are dependent from an allowable base claim 10. Claims 24 and 26-31 are allowable at least because they are dependent from an allowable base claim 23.

Conclusion:

In view of the above, the applicants submit that all of the pending claims are now allowable. Allowance at an early date would be greatly appreciated. Should any outstanding issues remain, the examiner is encouraged to contact the undersigned at (408) 293-9000 so that any such issues can be expeditiously resolved.

Respectfully Submitted,

Dated: June 7, 2006

Derek J. Westberg (Reg. No. 40,872)